qh

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/849,736	05/04/2001	G. Glenn Henry	CNTR:2021	8084

23669 7590 11/28/2005

HUFFMAN LAW GROUP, P.C.
1832 N. CASCADE AVE.
COLORADO SPRINGS, CO 80907-7449

EXAMINER	
HUISMAN, DAVID J	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 11/28/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/849,736	HENRY ET AL.	
	Examiner	Art Unit	
	David J. Huisman	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *03 October 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *31,34-36,38 and 43-49* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *31,34-36,38 and 43-49* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *7/11/05 & 10/20/05*.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1. Claims 31, 34-36, 38, and 43-49 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: IDS as received on 7/11/2005, Amendment as received on 10/3/2005, and IDS as received on 10/20/2005.

Information Disclosure Statement

3. All references disclosed by applicant on 7/11/2005 and on 10/20/2005 have been considered except for Hoyt, U.S. Patent No. 2-18-1997, which had been cited by the examiner in the previous Office Action. A line is been drawn through Hoyt and all other references have been initialed.

Maintained Rejections

4. Applicant has failed to overcome the prior art rejections set forth in the previous Office Action for all remaining claims. Consequently, these rejections are respectfully maintained by the examiner and are copied below for applicant's convenience.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 31, 43-44, and 46-49 are rejected under 35 U.S.C. 103(a) as being unpatentable over Narayan et al., U.S. Patent No. 5,850,532 (herein referred to as Narayan) in view of Hoyt et al., U.S. Patent No. 5,604,877 (herein referred to as Hoyt).

7. Referring to claim 31, Narayan has taught a pipelined microprocessor comprising:

a) an instruction cache (Fig.1, component 16) that is indexed by a fetch address, said instruction cache for caching instructions, wherein said instructions comprise variable byte-length instructions. See column 4, lines 36-37.

b) Narayan has not explicitly taught providing said instructions to an instruction buffer for storage therein. However, Official Notice is taken that an instruction buffer, such as a prefetch buffer, is well known and accepted in the art. More specifically, a prefetch buffer prefetches and stores instructions from the instruction cache, thereby making instructions available to the processor immediately, as opposed to the processor having to perform a somewhat time-expensive fetch from cache memory. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Narayan to include a buffer which receives and stores instructions from the instruction cache.

c) a branch target address cache (Fig.1, component 14), coupled to said instruction buffer and indexed by said fetch address, for caching branch target addresses of previously executed branch instructions. See column 7, lines 43-46.

d) said instruction buffer comprising an indicator associated with each byte of each of said instructions stored in said instruction buffer, wherein said indicator has a true value if said

branch target address cache predicts that said byte is an opcode byte of one of said instructions. See column 2, lines 41-43, and note a functional bit's setting determines whether the corresponding byte is an opcode byte.

e) Narayan has not taught that the indicator also indicates that said one of said instructions is one of said previously executed branch instructions and the microprocessor has speculatively branched to one of said branch target addresses cached for said one of said previously executed branch instructions. However, Hoyt has taught such a concept. See column 12, lines 58-64. Note that an indication is used to indicate that a prediction was made for a branch. Hoyt implements such an indication to track which instructions the system believes are branched. The belief is then verified (or not verified) by additional logic. See column 13, lines 7-10. Such a system allows for early branch prediction, before the branch is decoded so that instructions may continue to be fetched immediately. As a result, in order to fetch instructions along a predicted path sooner, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Narayan to have such an indication.

8. Referring to claim 43, Narayan in view of Hoyt has taught a microprocessor as described in claim 31.

a) Narayan has further taught instruction decode logic, coupled to said instruction buffer, for decoding said instructions to indicate which byte of each of said instructions is an opcode byte. See Fig.1, component 12.

b) Narayan in view of Hoyt has not taught prediction check logic, coupled to receive said indicator associated with each byte of said instruction from said instruction buffer, wherein if one of said indicators associated with one of said instructions specifies the microprocessor has

speculatively branched to said one of said branch target addresses of said one of said instructions, said prediction check logic indicates that the microprocessor erroneously branched to said one of said branch target addresses if said byte of said one of said instructions associated with said one of said indicators is not indicated by said instruction decode logic to be said opcode byte. However, Official Notice is taken that it is common for caches (BTAC/BTB included) to be constructed such that multiple fetch addresses map to the same cache entry (technique referred to as aliasing). If in the situation where a non-opcode byte is encountered and it maps to a branch entry that is allocated to a branch opcode, then a prediction should not be made because it is unknown whether this non-opcode byte is actually associated with a branch instruction. But, since it maps to the same location as a branch, the system will assume a branch has been encountered and an erroneous prediction would have been made. Such a cache construction is desired because it is not often that instructions map to the same location and it also keeps the cache smaller. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to reduce the size of the BTAC and determine erroneous predictions if a non-opcode byte results in a prediction being made.

9. Referring to claim 44, Narayan in view of Hoyt has taught a microprocessor as described in claim 31.

a) Narayan has further taught instruction decode logic, coupled to said instruction buffer, for decoding said instructions to indicate which byte of each of said instructions is an opcode byte. See Fig.1, component 12.

b) Hoyt has further taught prediction check logic, coupled to receive said indicator associated with each byte of said instructions from said instruction buffer, wherein if one of said indicators

associated with one of said instructions specifies the microprocessor has speculatively branched to said one of said branch target addresses of said one of said instructions, said prediction check logic indicates that the microprocessor erroneously branched to said one of said branch target addresses if said instruction decode logic indicates said one of said instructions is a non-branch instruction. See column 13, lines 7-19.

10. Referring to claim 46, Narayan in view of Hoyt has taught a microprocessor as described in claim 31. Furthermore, Narayan in view of Hoyt has taught prediction check logic, coupled to receive said indicator associated with each byte of said instructions from said instruction buffer, wherein said one of said instructions is a branch instruction, wherein if one of said indicators associated with said branch instruction specifies the microprocessor has speculatively branched to said one of said branch target addresses of said branch instruction, said prediction check logic indicates that the microprocessor erroneously branched to said one of said branch target addresses if a resolved direction of said branch instruction does not match a direction of said branch instruction predicted by said branch target address cache. This is deemed to be inherent because if the resolved direction of a branch does not match the predicted direction, then a misprediction has occurred (erroneously branched).

11. Referring to claim 47, Narayan in view of Hoyt has taught a microprocessor as described in claim 31. Narayan in view of Hoyt has further taught prediction check logic, coupled to receive said indicator associated with each byte of said instructions from said instruction buffer, wherein said one of said instructions is a branch instruction, wherein if one of said indicators associated with said branch instruction specifies the microprocessor has speculatively branched to said one of said branch target addresses of said branch instruction, said prediction check logic

indicates that the microprocessor erroneously branched to said one of said branch target addresses if a resolved target address of said branch instruction does not match said one of said branch target addresses to which the microprocessor speculatively branched. This is deemed to be inherent because if the resolved target address of a branch does not match the speculative target address, then a misprediction has occurred (erroneously branched).

12. Referring to claim 48, Narayan in view of Hoyt has taught a microprocessor as described in claim 31. Hoyt has further taught a non-speculative branch predictor (Fig.5, component 51), coupled to said instruction buffer, for generating a non-speculative predicted target address of a branch instruction for which said branch target address cache provided said one of said branch target addresses to which the microprocessor speculatively branched, and branch control logic, coupled to receive said indicator associated with each byte of said instructions from said instruction buffer, wherein if one of said indicators associated with said branch instruction specifies the microprocessor has speculatively branched to said one of said branch target addresses of said branch instruction, said branch control logic causes the microprocessor to branch to said non-speculative predicted target address if said non-speculative predicted target address generated by said non-speculative branch predictor does not match said one of said branch target addresses of said branch instruction provided by said branch target address cache. See column 13, lines 45-67.

13. Referring to claim 49, Narayan in view of Hoyt has taught a microprocessor as described in claim 31. Hoyt has further taught a non-speculative branch predictor, coupled to said instruction buffer, for generating a non-speculative predicted direction of a branch instruction for which said branch target address cache provided said one of said branch target addresses to

which the microprocessor speculatively branched, and branch control logic, coupled to receive said indicator associated with each byte of said instructions from said instruction buffer, wherein if one of said indicators associated with said branch instruction specifies the microprocessor has speculatively branched to said one of said branch target addresses of said branch instruction, said branch control logic causes the microprocessor to branch to a next instruction sequential to said branch instruction if said non-speculative predicted direction generated by said non-speculative branch predictor is a not taken prediction. See column 13, lines 45-67.

14. Claims 34-36 and 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shiell et al., U.S. Patent No. 5,850,543 (as applied in the previous Office Action and herein referred to as Shiell) in view of Roberts et al., U.S. Patent No. 5,752,069 (herein referred to as Roberts), and further in view of Narayan and Hoyt, as applied above.

15. Referring to claim 34, Shiell has taught a method of speculatively branching in a pipelined microprocessor, comprising:

a) caching, in a branch target address cache (BTAC), a plurality of branch target addresses of previously executed branch instructions. See Fig.2, BTB 56 and column 7, lines 40-42.

b) Shiell has not taught caching a bit associated with each of said branch instructions, wherein said bit is true only if the associated branch instruction spans more than one instruction cache line. However, Roberts has taught a single bit associated with each instruction which is only true if the instructions spans multiple cache lines. See bit 1 of the ICMROM signal (column 30, lines 27-30). Such an indication informs the system that it must fetch the remaining portion of an instruction from the next cache line. By allowing instructions to span multiple lines, cache

storage would be utilized more efficiently. For instance, if each cache line holds 4 bytes, and a subset of instructions are 3 bytes in length, then if there is no wrapping, one byte of each line would be wasted (since you can't fit a second instruction in a line). However, with wrapping, that final byte of storage is utilized every time. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Shiell to cache an indication of whether an instruction spans an instruction cache line.

c) wherein the microprocessor processes variable byte-length instructions. See column 6, lines 32-37.

d) accessing said BTAC with a fetch address of an instruction cache after said caching (fig. 2 indicates that the fetch address FA is used to access the BTB 56; col. 8, lines 13-16 indicate that the accessing is done when branch history is stored in the BTB).

e) determining whether said fetch address hits in said BTAC in response to said accessing (col. 8, lines 13-16).

f) branching the microprocessor to one of said plurality of branch target addresses selected by said fetch address if said fetch address hits in said BTAC whether or not a branch instruction is cached in a line of said instruction cache indexed by said fetch address (col. 7, lines 40-45 indicate that the target address is used to generate the fetch address during the fetch stage before the decoding the instruction without knowledge of whether the instruction at that fetch address in the instruction cache is still a branch or not i.e. speculative execution [col. 8, lines 13-16] because the BTB and the instruction cache are accessed in parallel using the fetch address FA).

g) Shiell has further taught storing in an instruction buffer instructions provided by said instruction cache selected by said fetch address. See Fig.2, component 60.

h) Shiell has not taught storing a discrete indication for each byte of said instruction, wherein said discrete indication is true if said BTAC predicts said byte is an opcode byte of said instruction. However, Narayan has taught indicating an opcode byte. See column 2, lines 41-43, and note a functional bit's setting determines whether the corresponding byte is an opcode byte. Clearly, the system should have knowledge of where the opcode of an instruction is, otherwise, it would not be able to determine the type of instruction. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Shiell such that Shiell includes an indication as to the location of opcode bytes.

i) Shiell and Narayan have not taught that the indication is true if said branching the microprocessor to one of said plurality of branch target addresses was performed for said instruction. However, Hoyt has taught such a concept. See column 12, lines 58-64. Note that an indication is used to indicate that a prediction was made for a branch. Hoyt implements such an indication to track which instructions the system believes are branched. The belief is then verified (or not verified) by additional logic. See column 13, lines 7-10. Such a system allows for early branch prediction, before the branch is decoded so that instructions may continue to be fetched immediately. As a result, in order to fetch instructions along a predicted path sooner, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Shiell to have such an indication.

16. Referring to claim 35, Shiell in view of Roberts in view of Narayan and further in view of Hoyt has taught a method as described in claim 34. Shiell has further taught storing a branch direction prediction associated with each of said plurality of branch target addresses prior to said accessing said BTAC (fig. 3, "HIS_n" field; col. 8, lines 57-67).

17. Referring to claim 36, Shiell in view of Roberts in view of Narayan and further in view of Hoyt has taught a method as described in claim 35. Shiell has further taught that said branching the microprocessor to said one of said plurality of branch target addresses selected by said fetch address is performed only if said associated branch direction prediction indicates said branch instruction will be taken (Although not explicitly mentioned, the limitation is deemed inherent to the correct functioning of the method because the purpose of the prediction information when indicating that the speculative branch is taken is for instructing the processor to branch to the target address of the branch and not the next sequential address).

18. Referring to claim 38, Shiell in view of Roberts in view of Narayan and further in view of Hoyt has taught a method as described in claim 39. Hoyt has further taught determining from said discrete indication, subsequent to said storing, that said branching was performed. See column 13, lines 16-19.

19. Claim 45 is rejected under 35 U.S.C. 103(a) as being unpatentable over Narayan in view of Hoyt, as applied above, and further in view of Stiles, U.S. Patent No. 5,513,330 (as applied in the previous Office Action).

20. Referring to claim 45, Narayan in view of Hoyt has taught a microprocessor as described in claim 31.

a) Narayan has further taught instruction decode logic, coupled to said instruction buffer. See Fig.1.

b) Black has not taught that the BTAC is configured to cache a length of each of said previously executed branch instructions, the decode logic determines a length of each of said instructions,

and prediction check logic, coupled to receive said indicator associated with each byte of said instructions from said instruction buffer, wherein if one of said indicators associated with one of said instructions specifies the microprocessor has speculatively branched to said one of said branch target addresses of said one of said previously executed branch instructions, said prediction check logic indicates that the microprocessor erroneously branched to said one of said branch target addresses if said length received from said instruction decode logic does not match said length of said one of said previously executed branch instructions provided by said branch target address cache. However, Stiles has taught a branch prediction cache which supplies target instruction lengths so that alignment and fetching may be done efficiently. See the first paragraph in the "Summary of Invention" section. Stiles has also taught calculating the lengths of target instructions which are not yet registered in the branch cache. See Fig.2 and Fig.4. Although Stiles has not taught comparing the predicted and calculated addresses, a person of ordinary skill in the art would have recognized that the comparison would be useful in systems which experience aliasing (where multiple branches map to the same entry in cache). Basically, aliasing is sometimes allowed so that the cache may be kept smaller (there will not be an individual entry for each instruction). The hope is that it isn't often that instructions map to the same location, and so the reduction in cache size is worth the occasional penalty of an instruction mapping to a shared location. As a result it would have been obvious to one of ordinary skill in the art to implement aliasing in Narayan in view of Hoyt. However, with aliasing implemented, a branch might map to an address which belongs to another branch, and consequently, the target instructions and their lengths may be incorrect. Therefore, a misprediction can be determined by comparing the predicted addresses and the calculated addresses for the actual instructions.

Therefore, in order to detect mispredictions in a system with aliasing, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Narayan in view of Hoyt to include a comparison of predicted and actual instruction lengths.

Response to Arguments

21. Applicant's arguments filed on October 3, 2005, have been fully considered but they are not persuasive.

22. Applicant argues the novelty/rejection of claim 31 on page 12 of the remarks, in substance that:

> "Notably, Applicant can find no teaching in Narayan of the functional bits indicating that Narayan's branch prediction unit predicted a byte is an opcode byte of an instruction in an instruction buffer. Rather, the predecode unit generates the functional bits. Narayan does state that the predecode unit may be configured to detect branch instructions and to store branch prediction information corresponding to the branch instructions into the branch prediction unit. However, Narayan provides no further detail on the nature of the branch prediction information or how it is used, and in particular, states nothing about the branch prediction information indicating Narayan's branch prediction unit predicted that a byte is an opcode byte of an instruction. See col. 6, lines 54-58."

23. These arguments are not found persuasive for the following reasons:

a) The examiner believes that applicant is reading the claim too narrowly. The claim language states that an instruction buffer comprises "an indicator associated with each byte of each of said instructions stored in said instruction buffer, wherein said indicator has a true value if said branch target address cache predicts that said byte is an opcode byte of one of said instructions." It should be noted that this language does not necessarily have to mean that the indicator is set to true subsequent to the branch cache predicting that the byte is an opcode byte, as applicant appears to be arguing. Instead, the claim is being interpreted in a more broad fashion. Namely, if a branch prediction is made for a particular byte, then the branch predictor believes (predicts)

that it has encountered a branch instruction (i.e., a branch opcode). The only type of byte that can indicate the type of instruction is the opcode byte. Since the branch predictor only predicts for branch opcodes, the byte indicator must be true. That is, the byte indicator must be true, i.e., the byte must be an opcode, for the predictor to make a prediction. The predictor will not make a prediction for a non-opcode byte because non-opcode bytes do not indicate the presence of branch instructions. The examiner believes that applicant needs to modify the claim to make it more clear that the indication is <u>set to true in response</u> to the branch cache predicting that said byte is an opcode byte. Right now, even if the indicator is set before the prediction, as in Narayan, it still reads on the claim. In amending, however, applicant should realize that Hoyt also does teach marking an instruction as being predicted (indicator is set to true which indicates that a branch opcode has been encountered), so Hoyt should be overcome as well.

Conclusion

24. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (571) 272-4168. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

DJH
David J. Huisman
November 19, 2005

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100